

# Author: Dibrova Volodymyr

## Practical Task 1: Linux Virtual Machine Setup and NSG Configuration

Create and configure a Linux Virtual Machine (VM) on Azure and secure it with a Network Security Group (NSG).

### Requirements:

1. Create a Linux VM (Ubuntu or CentOS) in Azure using the free tier.
2. Connect to the VM via SSH using a public-private key pair.
3. Install and configure an Nginx web server on the VM.
4. Create and configure a Network Security Group (NSG) to allow only HTTP (port 80) and SSH (port 22) traffic.
5. Test access to the Nginx web server from a browser.
6. Verify that any other ports are blocked by the NSG.

### Actions Taken:

1. Created a Linux VM (Ubuntu LTS)
2. Connected successfully using the command:

```
ssh azureuser@172.201.182.152
```

The screenshot shows the Azure portal interface with the 'Virtual machines' section. A single VM named 'MyLinuxVM' is listed. To the right, a Windows PowerShell window is open, showing the command sequence to create a VM and the resulting JSON output. Below the JSON, the terminal shows the successful SSH connection to the VM.

```
support to increase the quota for resource type PublicIpAddress", "details":[]}}}
PS C:\Users\User>
PS C:\Users\User> az vm create \
>> --resource-group Lesson2ComputeService \
>> --name MyLinuxVM \
>> --image Ubuntu2004 \
>> --admin-username azureuser \
>> --generate-ssh-keys
Selecting "usouth" may reduce your costs. The region you've selected may cost more.
To change this message in the future with the command "az config set core.display_region_id ://go.microsoft.com/fwlink/?linkid=222871
{
  "fqdns": "",
  "id": "/subscriptions/507cbe71-1145-4a77-bdb8-d6fa9921aed5/resourceGroups/Lesson2ComputeService/providers/virtualMachines/MyLinuxVM",
  "location": "westeurope",
  "macAddress": "60-45-BD-92-83-50",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "172.201.182.152",
  "resourceGroup": "Lesson2ComputeService",
  "zones": ""
}
PS C:\Users\User> ssh azureuser@172.201.182.152
The authenticity of host '172.201.182.152 (172.201.182.152)' can't be established.
ED25519 key fingerprint is SHA256:YunwGZSFtuK6FNTu2ns0+T4k599GgIHEOYn0qS/cRVU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.201.182.152' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1020-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro
```

### 3. Installed and Configured Nginx:

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install nginx -y
```

```
sudo systemctl start nginx
```

```
sudo systemctl status nginx
```

#### 4. Configured a Network Security Group (NSG):

Created rules to allow:

- **SSH (port 22)** for remote management.
  - **HTTP (port 80)** for web server access.

**Inbound Security Rules**

Priority	Name	Port	Protocol	Source	Destination	Action
100	HTTP-Allow	80	TCP	Any	Any	Allow
110	SSH-Allow	22	TCP	Any	Any	Allow
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	Allow

```

azureuser@MyLinuxVM:~$ curl http://172.201.182.152
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

< > C ⚠ Not secure 172.201.182.152

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## 5.Verified NSG Security using nmap:

```

azureuser@MyLinuxVM:~$ nmap 172.201.182.152
Starting Nmap 7.80 ( https://nmap.org ) at 2025-01-10 14:54 UTC
Nmap scan report for 172.201.182.152
Host is up (0.0047s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 4.89 seconds
Verified that only ports 22 (SSH) and 80 (HTTP) were open, while all other
ports were blocked.

```

### Practical Task 2: Windows Virtual Machine and RDP Access Setup

Set up a Windows Virtual Machine (VM) on Azure and configure access via Remote Desktop Protocol (RDP).

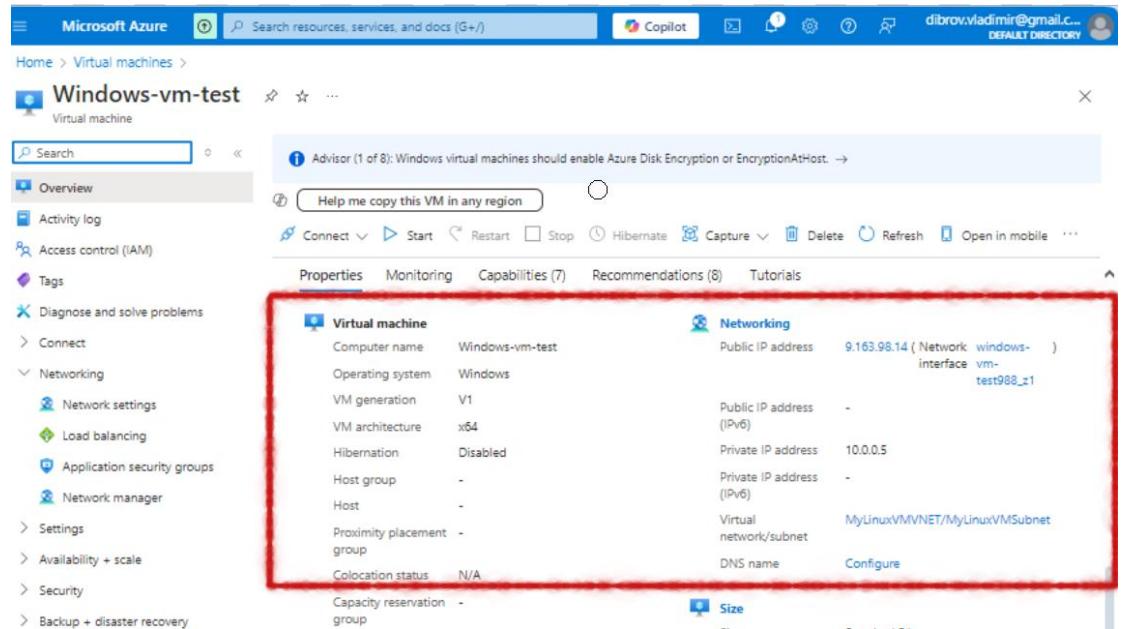
#### Requirements:

1. Create a Windows VM (e.g., Windows Server 2019) in Azure using the free tier.
2. Enable and configure Remote Desktop Protocol (RDP) for secure access to the VM.

3. Connect to the VM via RDP using Azure credentials.
4. Install a web server role (IIS) and deploy a simple test HTML page.
5. Verify access to the test page from a browser.
6. Ensure that unnecessary ports are closed, allowing only RDP (port 3389) and HTTP (port 80).

**Actions Taken:**

### **Created IIS ON Windows VM:**



The screenshot shows the Azure portal interface for managing a virtual machine. The main title bar says "Microsoft Azure" and "dibrov.vladimir@gmail.com". The left sidebar lists navigation options like "Home", "Virtual machines", "Windows-vm-test" (selected), "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Connect", "Networking", "Network settings", "Load balancing", "Application security groups", "Network manager", "Settings", "Availability + scale", "Security", and "Backup + disaster recovery". The main content area shows the "Properties" tab selected. A red box highlights the "Networking" section, which displays the following details:

Virtual machine	Networking
Computer name: Windows-vm-test	Public IP address: 9.163.98.14 (Network interface windows-vm-test988_z1)
Operating system: Windows	Public IP address (IPv6): -
VM generation: V1	Private IP address (IPv6): 10.0.0.5
VM architecture: x64	Private IP address (IPv6): -
Hibernation: Disabled	Virtual network/subnet: MyLinuxVMNET/MyLinuxVMSubnet
Host group: -	DNS name: Configure
Host: -	
Proximity placement group: -	
Colocation status: N/A	
Capacity reservation group: -	

## Configured Network Security Group (NSG):

The screenshot shows the Azure portal interface for managing a Network Security Group (NSG). The NSG is named "Windows-vm-test-nsq". The "Inbound Security Rules" section is highlighted with a red box. It contains three rules:

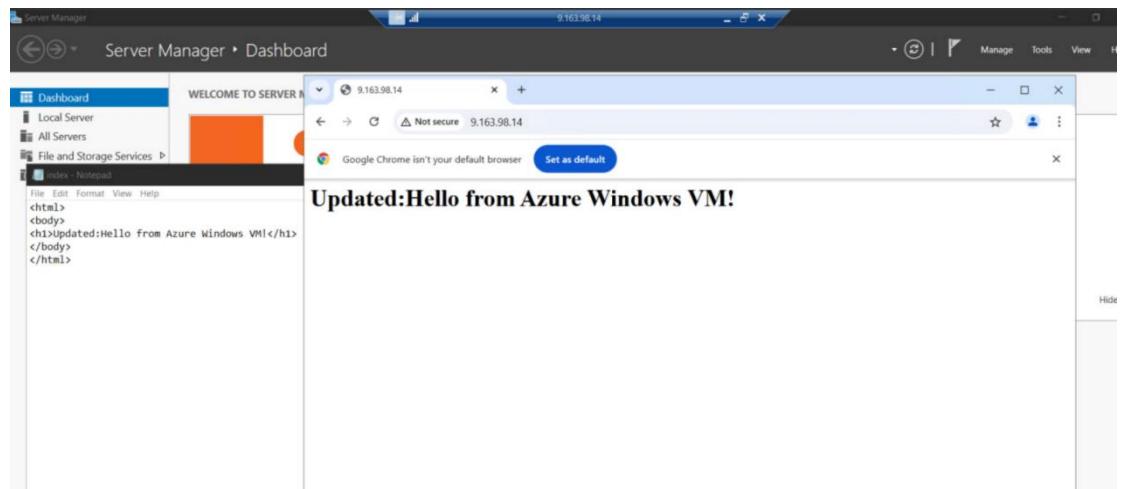
Priority	Name	Port	Protocol	Source	Destination	Action
1010	RDC	3389	TCP	Any	Any	Allow
1020	http	80	TCP	Any	Any	Allow
1030	https	443	TCP	Any	Any	Allow

## Connected to the VM via RDP and installed IIS Web Server:

1. Enabled the IIS role through **Server Manager**.
2. Deployed a custom `index.html` file to display a test message.

The screenshot shows the Windows Server interface. At the top, the IIS homepage is displayed, featuring a grid of "Welcome" messages in various languages. Below it, the Server Manager dashboard shows performance metrics like CPU Usage and Available Memory. The "Local Server" section is selected in the navigation pane. The "ROLES AND FEATURES" section at the bottom lists the installed Web Server (IIS) role.

Server Name	Name	Type	Path
Windows-vm-test	Web Server (IIS)	Role	Web Server (IIS)



## Verified Security Configuration:

- Used **nmap** to ensure only ports 3389 and 80 were open.

```
PS C:\Users\User> nmap -Pn 9.163.98.14
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-11 22:33 FLE Standard Time
Nmap scan report for 9.163.98.14
Host is up (0.065s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 14.25 seconds
PS C:\Users\User> |
```

## Practical Task 3: Configuring an Azure Load Balancer

Create and configure a **Basic Azure Load Balancer** to distribute traffic across multiple virtual machines.

### Requirements:

- Create two Linux or Windows virtual machines in the same region and virtual network using the Azure Free Tier.
- Install and configure a web server (e.g., Nginx on Linux or IIS on Windows) on both VMs with unique content for testing.
- Create a **Basic Load Balancer** in Azure (included in the free tier) and configure it to balance HTTP (port 80) traffic between the two virtual machines.
- Configure a health probe to monitor the availability of the VMs.
- Test the Load Balancer by accessing its public IP address from a browser and verify that traffic is routed to both VMs (by observing the unique content from each server).
- Verify that the Load Balancer removes unavailable VM from the traffic pool when it fails the health probe.

## Actions Taken:

### Created Two Virtual Machines Windows and Linux :

The screenshot shows the Microsoft Azure portal interface. The user is in the 'Virtual machines' section under the 'Compute' service. There are two VMs listed:

Name	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disk
MyLinuxVM	Azure subscription 1	Lesson2ComputeService	West Europe	Running	Linux	Standard_DS1_v2	172.201.182.152	1
Windows-vm-test	Azure subscription 1	Lesson2ComputeService	West Europe	Stopped (deallocated)	Windows	Standard_B1s	9165.98.14	1

### Configured a Basic Load Balancer:

1. Set up a **Frontend IP** with a public IP address.
2. Added both VMs to the **Backend Pool** to handle incoming traffic.

The screenshot shows the Microsoft Azure portal interface. The user is in the 'Load balancing' section under the 'Compute' service. They are viewing the 'Frontend IP configuration' for a load balancer named 'MyBasicLB'. One frontend IP is listed:

FrontendIP	IP address	Rules count
100.142.22.234 (MyPublicIP)	100.142.22.234 (MyPublicIP)	1

The screenshot shows the Microsoft Azure portal interface. The user is in the 'Load balancing' section under the 'Compute' service. They are viewing the 'Backend pools' for the 'MyBasicLB' load balancer. Two backend pools are listed:

Backend pool	Resource Name	IP address	Network interface	Availability zone	Rules count	Resource Status	Admin state
MyLinuxBackendPool	MyLinuxVM	10.0.0.4	MyLinuxVMVMNIC	-	1	Stopped (deallocated)	None
MyWindowsBackendPool	Windows-vm-test	10.0.0.5	window-vm-test988.z1	1	1	Running	None

## Set Up a Health Probe:

1. Configured an HTTP Health Probe to monitor the availability of the VMs on port 80.

The screenshot shows two overlapping Azure portal pages. The top page is titled 'MyBasicLB | Health probes' and displays a table of existing health probes. One probe, 'HTTPProbe', is listed with 'Http' as the protocol, port 80, and path '/'. The bottom page is titled 'HTTPRule' and shows the configuration for a new rule. The 'Name' field is set to 'HTTPRule'. Under 'Frontend IP address', 'FrontendIP1 (108.142.28.234)' is selected. The 'Backend pool' is set to 'MyLinuxBackendPool'. The 'Protocol' is set to 'TCP'. The 'Port' and 'Backend port' are both set to 80. The 'Health probe' dropdown shows 'HTTPProbe (HTTP-80)' selected. The 'Session persistence' dropdown is set to 'None'. The 'Idle timeout (minutes)' is set to 4. Under 'Outbound source network address translation (SNAT)', the 'Use outbound rules to provide backend pool members access to the internet' option is selected. Both 'Enable TCP Reset' and 'Enable Floating IP' checkboxes are unchecked.

## Verified Traffic Distribution:

1. Accessed the Load Balancer's public IP address through a browser.
2. Stopped one VM (Windows VM) to test Load Balancer behavior.
3. Verified that the Load Balancer excluded the unavailable VM and routed traffic only to the healthy Linux VM.

The screenshot shows the Azure portal interface with two main windows open:

- Virtual machines**: A list of two VMs:
 

Name	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address
MyLinuxVM	Azure subscription 1	Lesson2ComputeSeri...	West Europe	Running	Linux	Standard_DS1_v2	172.201.182.1
Windows-vm-test	Azure subscription 1	Lesson2ComputeSeri...	West Europe	Stopped (deallocated)	Windows	Standard_B1s	9.163.98.14
- MyBasicLB - Microsoft Azure**: Load Balancer settings:
 

Name	IP address	Rules count
FrontendP1	108.142.28.234 (MyPublicIP)	1

Both the VM list and the load balancer configuration window have red boxes highlighting specific entries: the status of the Linux VM, its operating system, and its public IP address; and the IP address of the frontend pool.

**Practical Task 4: Configuring a Basic Load Balancer with Virtual Machine Scale Sets (VMSS)**  
**Set up a Basic Azure Load Balancer to distribute traffic across a Virtual Machine Scale Set (VMSS).**  
**Requirements:**

1. Create a Virtual Machine Scale Set (VMSS) in Azure using Linux or Windows instances within the free tier (for example **B1s size**). Limit the scale set to two VM instances to avoid exceeding the free-tier 750-hour limit.
2. Deploy the scale set with a custom configuration to install and configure a web server (e.g., Nginx on Linux or IIS on Windows) on each VM instance.
3. Configure the **Basic Load Balancer** to distribute HTTP (port 80) traffic across the VM instances in the scale set.
4. Add a health probe to monitor the availability of instances in the VMSS.
5. Scale the VMSS manually by increasing the number of instances to verify the Load Balancer routes traffic to the newly added VMs.
6. Test the setup by accessing the Load Balancer's public IP address and verifying traffic distribution across multiple VM instances.
7. Verify that the Load Balancer removes an unavailable instance from the traffic pool when it fails the health probe.

## Actions Taken:

### Created Virtual Machine Scale Set (VMSS):

The screenshot shows the Azure portal interface for managing a Virtual Machine Scale Set (VMSS). The URL is <https://portal.azure.com/#@dibrovvladimir@gmail.onmicrosoft.com/resource/subscriptions/507cbe71-1145-4a77-bdb8-d6fa9921aed5/resourceGroups/MyVMSSGroup/providers/Microsoft.Compute/virtualMachineScaleSets/MyWindowsVMSS>. The page title is "MyWindowsVMSS | Instances". The left sidebar shows the navigation path: Home > Virtual machine scale sets > MyWindowsVMSS. The main content area displays the "Instances" section of the VMSS. There are two instances listed: "MyWindowsVMSS\_0" and "MyWindowsVMSS\_1". Both instances are shown as green dots, indicating they are running. The table columns include Instance, Computer name, Status, Protection policy, Provisioning state, Health state, and Latest model. The "Status" column shows "Running" for both instances. The "Protection policy" column shows "Succeeded" for both. The "Provisioning state" column shows "Succeeded" for both. The "Health state" column shows "Yes" for both. The "Latest model" column shows "Yes" for both.

Instance	Computer name	Status	Protection policy	Provisioning state	Health state	Latest model
MyWindowsVMSS_0	mywindows000000	Running	Succeeded	Succeeded	Yes	Yes
MyWindowsVMSS_1	mywindows000001	Running	Succeeded	Succeeded	Yes	Yes

## Configured Load Balancer && Set Up Health Probe

**Microsoft Azure | Load balancing | Load Balancer | MyLoadBalancer**

**MyLoadBalancer | Frontend IP configuration**

Name	IP address	Rules count
MyLoadBalancer-frontendconfig01	172.172.230.194 (MyLoadBalancer-publicip)	1

**MyLoadBalancer-probe01**

Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Unhealthy threshold: otherwise known as the property `numberOfProbes`, is not respected. Load Balancer health probes will probe up/down immediately after 1 probe regardless of the property's configured value. [Learn more](#)

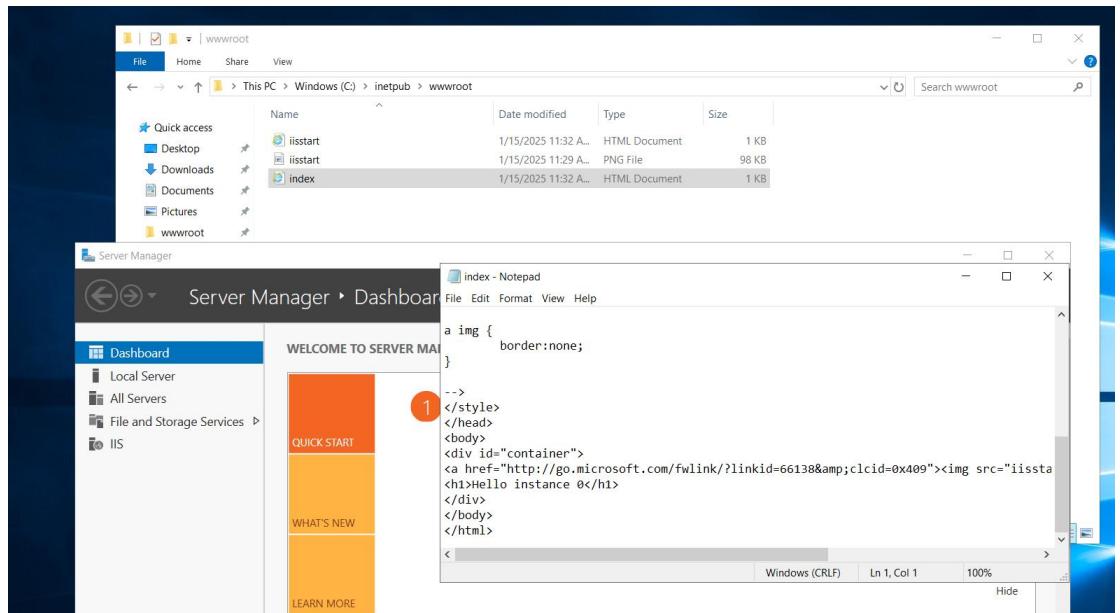
Name *	MyLoadBalancer-probe01
Protocol *	HTTP
Port *	80
Path *	/
Interval (seconds) *	15
Used by *	MyLoadBalancer-lrule01

**MyLoadBalancer | Load balancing rules**

Name	Protocol	Backend pool	Health probe	Health status
MyLoadBalancer-lrule01	TCP/80	bepool	MyLoadBalancer-probe01	<a href="#">View details</a>

**MyLoadBalancer | Backend pools**

Backend pool	Resource Name	IP address	Network interface	Availability zone	Rules count	Resource Status	Admin state
bepool (2)	MyWindowsVM5 (instance 0)	172.16.0.4	vnet-eastus-1-nic01	-	1	Running	None
bepool	MyWindowsVM5 (instance 1)	172.16.0.5	vnet-eastus-1-nic01	-	1	Running	None



This screenshot illustrates the deployment of a web application via Azure. On the left, the Azure portal shows the 'MyWindowsVMSS\_0' virtual machine scale set instance. It highlights the public IP address (172.174.80.243) and the private IP address (172.18.0.4). On the right, the IIS landing page is displayed, featuring a grid of international welcome messages and a central banner that says 'Hello instance 0'.

The screenshot displays two separate Azure environments. On the left, the Microsoft Azure portal shows the 'Instances' blade for a resource group named 'MyWindowsVMSS'. It lists a single instance named 'MyWindowsVMSS\_1' with a public IP address of 172.174.80.193. On the right, a browser window shows the 'Internet Information Services' home page for this instance, with the URL 172.174.80.193 highlighted in a red box. The page features a grid of international greetings and a central banner with the text 'Hello Instance 1'. A second browser window on the left shows the 'Instances' blade for another resource group named 'MyWindowsVMSS'. This instance has a public IP address of 172.172.230.194, which is also highlighted in a red box. Its corresponding IIS home page is shown on the right, with the URL 172.172.230.194 highlighted in a red box. This page also features a grid of international greetings and a central banner with the text 'Hello Instance 1'.

## 2 method:

### Helping with Custom Script Extension

This screenshot shows the 'Storage accounts > Containers' blade in the Microsoft Azure portal. A specific container named 'newiis' is selected. Below the container name, there are download links for 'Upload', 'Refresh', and 'Give feedback'. An 'Authentication method' dropdown is set to 'Access key (Switch to Microsoft Entra user account)'. The 'Location' field contains the value 'newiis'. A search bar at the bottom allows filtering blobs by prefix. A table below lists a single blob named 'iis.ps1' with details such as 'Modified' (1/16/2025, 1:18:15 PM), 'Access tier' (Standard), 'Archive status' (Not yet archived), 'Blob type' (Block blob), 'Size' (135 B), and 'Lease state' (Available).

```

C:\Users\User\Downloads\iis.ps1.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
iis.ps1.txt x
1 Install-WindowsFeature -name Web-Server -IncludeManagementTools
2 echo "Hello from Azure VMSS Instance!" > C:\inetpub\wwwroot\index.html
3

```

Home >

## Microsoft.CustomScriptExtension | Overview

Deployment

Search X < Delete Cancel Redeploy Download Refresh

**Overview**

- Inputs
- Outputs
- Template

Your deployment is complete

Deployment name : Microsoft.CustomScriptExtension  
Subscription : Azure subscription 1  
Resource group : MyVMSSGroup

Deployment details

Next steps

Go to resource

Scaling was not possible due to the limitations of the Azure Free Tier

MyWindowsVMSS | Scaling

Virtual machine scale set

There was an error updating instance count for resource 'MyWindowsVMSS'. Detail message :

```

{
  "error": {
    "code": "PublicIPCountExceededError",
    "message": "The requested number of publicIPAddresses 3 for VM Scale Set (subscriptionId=507cbe71-1145-4a77-bdb8-d0fa9921aed5/resourceGroups/MyVMSSGROUP/providers/Microsoft.Compute/virtualMachineScaleSets/MyWindowsVMSS will exceed the maximum number of publicIPAddresses allowed 3 for subscription 507cbe71-1145-4a77-bdb8-d0fa9921aed5."
  }
}

```

Please try again in a few moments.

Configure Scale-In Policy Predictive charts Run history JSON Notify Diagnostic settings

Autoscale is a built-in feature that helps applications perform their best when demand changes. You can choose to scale your resource manually to a specific instance count, or via a custom Autoscale policy that scales based on metric(s) thresholds, or schedule instance count which scales during designated time windows. Autoscale enables your resource to be performant and cost effective by adding and removing instances based on demand. Learn more about Azure Autoscale or [view the how-to video](#).

Choose how to scale your resource

- Manual scale
- Custom autoscale

Manual scale

Override condition

Instance count : 3

### Results:

- ✓ Successfully configured the Load Balancer to distribute traffic evenly across the initial two VMSS instances.

- ✓ Load Balancer correctly excluded the unavailable instance from the traffic pool when it failed the Health Probe.
- ✓ Scaling attempt failed due to Azure Free Tier usage limits, preventing the addition of a third B2 instance.
- ✓ All HTTP requests to the Load Balancer's public IP were routed to healthy VMSS instances

## Practical Task 5: Deploying a Web Application Using Azure App Services

Set up and deploy a simple web application using **Azure App Services**.

### Requirements:

1. Create an Azure App Service (Web App) using the Azure Free Tier.
2. Select the runtime stack of your choice (e.g., .NET, Python, Node.js) during the setup.
3. Develop or use a sample web application (e.g., a "Hello World" app) and deploy it to the App Service using:
  - o Azure Portal
  - o Azure CLI
  - o Or direct deployment from a GitHub repository.
4. Test the deployed application by accessing its URL provided by Azure App Services.
5. Enable **App Service Logs** and verify that application logs are being generated.
6. Clean up resources after completion to avoid unnecessary usage.

### Actions Taken:

#### Created an Azure App Service:

1. Set up an App Service using the Azure Free Tier.
2. Selected Java 11 as the runtime stack.

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type
JustApp	Stopped	West Europe	Free	ASP-MyVMSSGroup-91a6	Azure subscription 1	Web App

#### Developed a Simple Web Application:

1. Created a "Hello World" application using Spring Boot.
2. Pushed the application code to a github

The screenshot shows a GitHub code editor interface. At the top, there's a navigation bar with back, forward, and search icons, followed by the URL "github.com/Vivien87/mywebdemo/edit/main/mywebdemo/src/main/java/com/myweb/mywebdemo/HelloController.java". Below the URL are several small icons for different GitHub features like issues, pull requests, and settings.

The main area has a header "Vivien87 / mywebdemo" and a search bar "Type [ ] to search". Below the header are tabs for "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings".

The left sidebar is titled "Files" and shows a tree view of the project structure:

- main
  - mywebdemo
  - src
    - main
      - java/com/myweb/mywebde...
      - HelloController.java
      - MywebdemoApplication.java
    - resources
    - test/java/com/myweb/myweb...
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

The right pane displays the content of "HelloController.java". The code is as follows:

```

1 package com.myweb.mywebdemo;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HelloController {
8     @GetMapping("/")
9     public String hello() {
10         return "Hello, World from Azure App Services!";
11     }
12 }
13

```

A tooltip at the bottom of the right pane says: "Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use Esc than tab to move to the next interactive element".

## Deployed the Application:

1. Uploaded the application to Azure App Service through the Azure Portal.

The screenshot shows a Windows PowerShell window with the title "Windows PowerShell". The command entered is:

```
PS C:\Users\User> az webapp deployment source config `
```

The output shows the configuration of the Azure Web App deployment source:

```

>> --resource-group MyVMSSGroup `
>> --name JustApp `
>> --repo-url https://github.com/Vivien87/mywebdemo `
>> --branch main `
>> --manual-integration
location is not a known attribute of class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.SiteSourceControl'> and
will be ignored
{
    "branch": "main",
    "deploymentRollbackEnabled": false,
    "gitHubActionConfiguration": null,
    "id": "/subscriptions/507cbe71-1145-4a77-bdb8-d6fa9921aed5/resourceGroups/MyVMSSGroup/providers/Microsoft.Web/sites/JustApp/sourcecontrols/web",
    "isGitHubAction": false,
    "isManualIntegration": true,
    "isMercurial": false,
    "kind": null,
    "location": "West Europe",
    "name": "JustApp",
    "repoUrl": "https://github.com/Vivien87/mywebdemo",
    "resourceGroup": "MyVMSSGroup",
    "tags": {
        "hidden-link: /app-insights-conn-string": "InstrumentationKey=e41e90b5-ef22-4d29-b0b8-61ablece309;IngestionEndpoint=https://westeurope-5.in.applicationinsights.azure.com/;LiveEndpoint=https://westeurope.livediagnostics.monitor.azure.com/";
        "ApplicationId": "8a125406-42ec-4617-96d0-2d3da8e76d14",
        "hidden-link: /app-insights-instrumentation-key": "e41e90b5-ef22-4d29-b0b8-61ablece309",
        "hidden-link: /app-insights-resource-id": "/subscriptions/507cbe71-1145-4a77-bdb8-d6fa9921aed5/resourceGroups/MyVMSSGroup/providers/microsoft.insights/components/JustApp"
    },
}
```

2. Checking deployment status

```

PS C:\Users\User> az webapp log tail --name JustApp --resource-group MyVMSSGroup
2025-01-16T10:41:55 Welcome, you are now connected to log-streaming service. The default timeout is 2 hours. Change the
timeout with the App Setting SCM_LOGSTREAM_TIMEOUT (in seconds).
2025-01-16 10:41:52 JUSTAPP GET / - 443 - 10.0.128.4 ReadyForRequest/1.0+(LocalCache) - - justapp-e3aeessdh9asfbj.weste
urope-01.azurewebsites.net 200 0 0 366 669 1
2025-01-16 10:41:52 JUSTAPP GET / - 443 - 10.0.128.4 ReadyForRequest/1.0+(AppInit) - - justapp-e3aeessdh9asfbj.westeuro
pe-01.azurewebsites.net 200 0 0 363 633 16
2025-01-16 10:41:52 JUSTAPP GET / X-ARR-LOG-ID=15247e9b-c39b-4617-aab0-6f832f65bc41 443 - 95.158.42.217 python-urllib3/1
.26.19 - justapp-e3aeessdh9asfbj.westeurope-01.azurewebsites.net 200 0 0 415 890 22
2025-01-16 10:42:11 JUSTAPP GET / X-ARR-LOG-ID=12447d2c-6ade-413a-b0c4-0e8de69b6e8e 443 - 95.158.42.217 Mozilla/5.0+(Win
dows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/130.0.0.0+Safari/537.36+OPR/115.0.0.0 ARRAffinit
y=92ca53ad8db4fb93d4d3b7d8ab54dcf8ffecb2d731f25b0e91ad575d7534c3f;+ARRAffinitySameSite=92ca53ad8db4fb93d4d3b7d8ab54dcf
8ffecb2d731f25b0e91ad575d7534c3f https://sandbox-3.reactblade.portal.azure.net/ justapp-e3aeessdh9asfbj.westeurope-01.a
zurewebsites.net 200 0 0 577 1664 21
2025-01-16 10:42:15 JUSTAPP GET / X-ARR-LOG-ID=60c4f67a-c63d-427b-9da0-18719e915ce6 443 - 95.158.42.217 Mozilla/5.0+(Win
dows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/130.0.0.0+Safari/537.36+OPR/115.0.0.0 ARRAffinit
y=92ca53ad8db4fb93d4d3b7d8ab54dcf8ffecb2d731f25b0e91ad575d7534c3f;+ARRAffinitySameSite=92ca53ad8db4fb93d4d3b7d8ab54dcf
8ffecb2d731f25b0e91ad575d7534c3f https://sandbox-3.reactblade.portal.azure.net/ justapp-e3aeessdh9asfbj.westeurope-01.a
zurewebsites.net 200 0 0 577 1664 32
2025-01-16T10:43:55 No new trace in the past 1 min(s).
2025-01-16T10:44:55 No new trace in the past 2 min(s).
2025-01-16T10:45:55 No new trace in the past 3 min(s).
Terminate batch job (Y/N)?

```

My application is successfully deployed and running

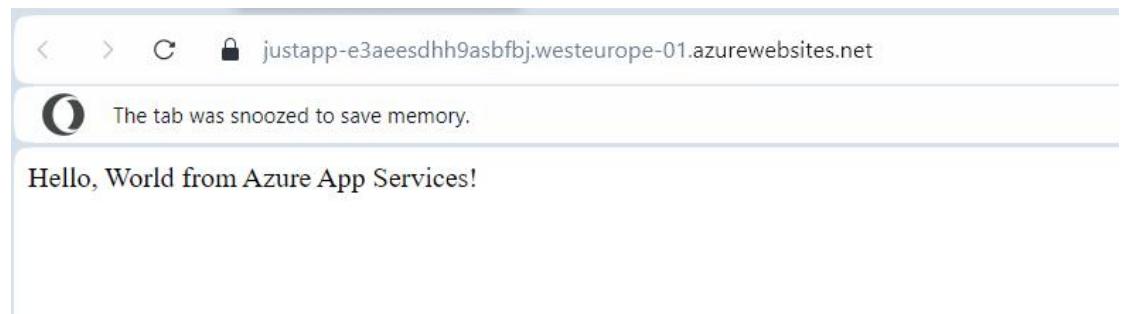
### 3. Getting the URL of my web application

```

PS C:\Users\User> az webapp browse --name JustApp --resource-group MyVMSSGroup
PS C:\Users\User> az webapp show --name JustApp --resource-group MyVMSSGroup --query defaultHostName --output tsv
justapp-e3aeessdh9asfbj.westeurope-01.azurewebsites.net
PS C:\Users\User>

```

### Tested the Application:



### Enabled Logs:

## 1. Turned on Application and Web Server Logs in the App Service settings.

The screenshot shows the Azure portal interface for managing an App Service named 'JustApp'. The left sidebar lists various service settings like Deployment, Monitoring, and Performance. The 'Monitoring' section is expanded, showing 'Logs' which is currently selected. On the right, there's a detailed configuration pane for 'App Service logs'. It includes sections for 'Application logging (Filesystem)' and 'Application logging (Blob)', both with 'On' buttons. The 'Logs' section displays a log stream with multiple entries, each showing a timestamp, source, and log message. At the bottom right, there's a note about 'No FTP/deployment user set'.

## 2. Monitored logs using the Log Stream feature to ensure proper functionality.

This screenshot shows the 'Log stream' feature for the 'JustApp' web app. The left sidebar shows the 'Log stream' option is selected. The main area displays a real-time log stream with numerous entries. Each entry contains a timestamp, a source identifier (like 'IIS-WebApp-0'), and a log message describing an event, such as a connection or a request being processed. The log stream is constantly updating, showing the dynamic nature of the application's logs.

## Cleaned Up Resources:

### 1. Deleted the App Service and related resources to avoid unnecessary usage

1 method:

```
az webapp delete --name JustApp --resource-group MyVMSSGroup
```

## 2 method:

The screenshot shows the Azure portal interface for managing app services. In the center, there's a detailed view of an app named 'JustApp'. On the left, a sidebar lists various management options like 'Activity log', 'Tags', and 'Deployment'. The main area has tabs for 'Properties', 'Monitoring', 'Logs', 'Capabilities', 'Notifications', and 'Recommendations'. On the right, there's a 'Deployment Center' section showing deployment logs and a success message from January 16 at 12:24:59 PM. The top navigation bar includes standard icons for search, copilot, and user info.

## Practical Task 6: Creating and Deploying an Azure Function to Process HTTP Requests

Set up and deploy an Azure Function that processes HTTP requests directly from the Azure portal.

### Requirements:

#### 1. Set Up the Function App:

- o Access the Azure portal and navigate to **Azure Functions**.
- o Create a new **Function App** using the Consumption (Serverless) plan within the free tier.

#### 2. Create a Function:

- o Select the **HTTP trigger** template.
- o Choose a language of your choice (e.g., C#, JavaScript, or Python).

#### 3. Customize the Function:

- o Modify the default code to return the following response:  
**Hello, [name]! Welcome to Azure Functions.**
- o If no name parameter is provided in the query string or request body, the response should be:  
**Hello! Please provide your name.**

#### 4. Test the Function:

- o Use the **Test/Run** feature in the Azure portal to send HTTP requests.
- o Verify the Function responds appropriately with and without the name parameter.

#### 5. Verify External Access:

- o Retrieve the public URL of the Azure Function.
- o Test the Function using a browser or a tool like Postman to ensure it's externally accessible.
- o Navigate to the **Monitor** tab of the Function App.
- o Check metrics like execution count, response time, and errors

## Actions Taken:

### Created Function App:

Microsoft Azure

JustFunctionApp

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Recommended services (preview)

Log stream

Functions

App keys

App files

Proxies

Deployment

Browse Refresh Stop Restart Swap Get publish profile Reset publish profile Download app content Delete Send us your feedback

Your app is not configured for dynamic scaling. Scaling could be limited. Click to learn more.

**Essentials**

Resource group (move)	: Lesson2ComputeService	Default domain	: justfunctionapp.azurewebsites.net
Status	: Running	Operating System	: Windows
Location (move)	: West Europe	App Service Plan	: ASP-Lesson2ComputeService-09ee (Y1:0)
Subscription (move)	: Azure subscription 1	Runtime version	: 4.1036.3.23284
Subscription ID	: 507dbe71-1145-4a77-bdb8-d6fa9921aed5		
Tags (edit)	: Add tags		

Functions Metrics Properties Notifications (1)

Set up local environment Refresh

Filter by name...

Name Trigger Status Monitor

HttpTriggerJava HTTP Enabled Invocations and more

## Created a Function:

1. Selected the **HTTP trigger** template. Chose **Java** as the development language.

AF azure-function Version control

Project C:\Users\User\Desktop\JustFunctionApp

pom.xml (azure-function) HttpTriggerJava.java

azure-function .idea src main java org.demo.functions HttpTriggerJava target Azure Deployment Targets JustFunctionApp Running Resource Connections .gitignore azure-function.iml hostison

Run Deploy Functions: azure-function

Starting deployment... Trying to deploy artifact to JustFunctionApp... Successfully deployed the artifact to <https://justfunctionapp.azurewebsites.net> Deployment succeed

Microsoft Azure

JustFunctionApp

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Recommended services (preview)

Log stream

Functions

App keys

App files

Proxies

Deployment

Settings

Performance

App Service plan

Browse Refresh Stop Restart Swap Get publish profile Reset publish profile Download app content Delete Send us your feedback

Your app is not configured for dynamic scaling. Scaling could be limited. Click to learn more.

**Essentials**

Resource group (move)	: Lesson2ComputeService	Default domain	: justfunctionapp.azurewebsites.net
Status	: Running	Operating System	: Windows
Location (move)	: West Europe	App Service Plan	: ASP-Lesson2ComputeService-09ee (Y1:0)
Subscription (move)	: Azure subscription 1	Runtime version	: 4.1036.3.23284
Subscription ID	: 507dbe71-1145-4a77-bdb8-d6fa9921aed5		
Tags (edit)	: Add tags		

Functions Metrics Properties Notifications (1)

Set up local environment Refresh

Filter by name...

Name Trigger Status Monitor

HttpTriggerJava HTTP Enabled Invocations and more

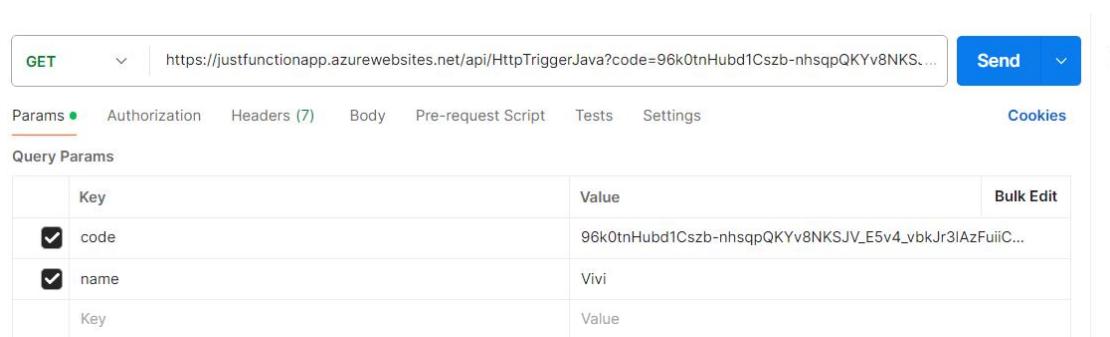
## Customized the Function:

## 1. Modified the default code to handle two scenarios:

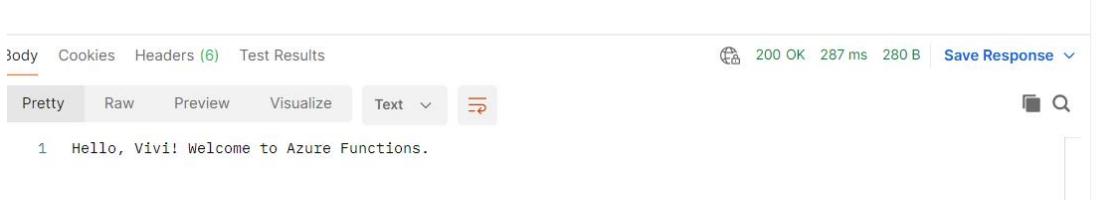
### 1. Returns "Hello, [name]! Welcome to Azure Functions."



The screenshot shows the Azure Functions Test/Run interface. The 'Output' tab is selected, displaying a 200 OK status and the response content: "Hello, Vivi! Welcome to Azure Functions."

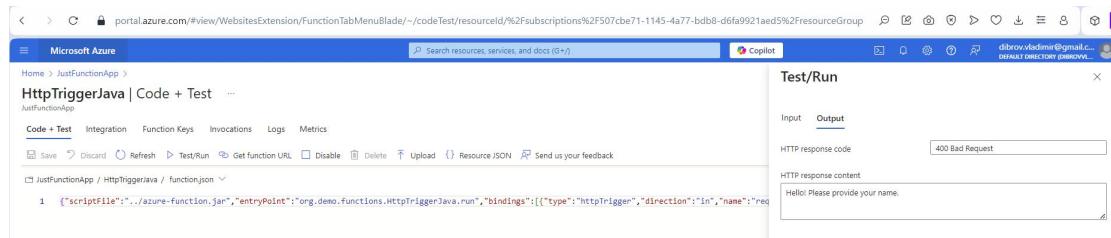


The screenshot shows a Postman request for a GET API. The URL is https://justfunctionapp.azurewebsites.net/api/HttpTriggerJava?code=96k0tnHubd1Cszb-nhsqpQKYv8NKS... . The 'Params' tab shows 'code' and 'name' as query parameters with values '96k0tnHubd1Cszb-nhsqpQKYv8NKSJV\_E5v4\_vbkJr3IAzFuijC...' and 'Vivi' respectively. The response body shows the message "Hello, Vivi! Welcome to Azure Functions."



The screenshot shows the Azure Functions Test/Run interface. The 'Output' tab is selected, displaying a 400 Bad Request status and the response content: "Hello! Please provide your name."

### 2. Returns "Hello! Please provide your name."



The screenshot shows the Azure Functions Test/Run interface. The 'Output' tab is selected, displaying a 400 Bad Request status and the response content: "Hello! Please provide your name."

GET https://justfunctionapp.azurewebsites.net/api/HttpTriggerJava?code=96k0tnHubd1Cszb-nhsqpQKYv8NKSJV\_E5v4\_vbkJ...

HTTP https://justfunctionapp.azurewebsites.net/api/HttpTriggerJava?code=96k0tnHubd1Cszb-nhsqpQKYv8NKSJV\_E5v4\_vbkJ...

GET https://justfunctionapp.azurewebsites.net/api/HttpTriggerJava?code=96k0tnHubd1Cszb-nhsqpQKYv8NKSJV\_E5v4\_vbkJ... Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
<input checked="" type="checkbox"/> code	96k0tnHubd1Cszb-nhsqpQKYv8NKSJV_E5v4_vbkJr3lAzFuiiC...	
<input type="checkbox"/> name	Vivi	
Key	Value	

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize Text

400 Bad Request 280 ms 234 B Save Response

Hello! Please provide your name.

## Monitor -Metrics:

